

PATENTTI- JA REKISTERIHALLITUS  
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 28.7.2003

ETUOIKEUSTODISTUS  
PRIORITY DOCUMENTHakija  
ApplicantNokia Corporation  
HelsinkiPatenttihakemus nro  
Patent application no

20021656

Tekemispäivä  
Filing date

16.09.2002

Kansainvälinen luokka  
International class

H03M

Keksinnön nimitys  
Title of invention**"Menetelmä ja järjestely dekodauksen suorittamiseksi"**

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä Patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.

  
Pirjo Kalla  
TutkimussihteeriMaksu 50 €  
Fee 50 EUR

Maksu perustuu kaupp- ja teollisuusministeriön antamaan asetukseen 1027/2001 Patentti- ja rekisterihallituksen maksullisista suoritteista muutoksineen.

The fee is based on the Decree with amendments of the Ministry of Trade and Industry No. 1027/2001 concerning the chargeable services of the National Board of Patents and Registration of Finland.

Osoite:	Arkadiankatu 6 A	Puhelin:	09 6939 500	Telefax:	09 6939 5328
	P.O.Box 1160	Telephone:	+ 358 9 6939 500	Telefax:	+ 358 9 6939 5328
	FIN-00101 Helsinki, FINLAND				

## Menetelmä ja järjestely dekodauksen suorittamiseksi

### Ala

Keksinnön kohteena on menetelmä ja järjestely Viterbi-dekodauksen suorittamiseksi. Erityisesti keksintö kohdistuu polkumetriikoiden laskennan toteuttamiseen.

### Tausta

Tietoliikennejärjestelmissä käytetty kanava aiheuttaa usein häiriötä tiedonsiirrolle. Häiriötä esiintyy kaikenlaisissa järjestelmissä, mutta erityisesti langattomissa tietoliikennejärjestelmissä siirtotie vaimentaa ja vääristää siirrettävää signaalia monin eri tavoin. Siirtotiellä häiriötä aiheuttavat tyypillisesti signaalin monitie-eteneminen, erilaiset häipymät ja heijastukset sekä myös toiset samalla siirtotiellä siirrettävät signaalit.

Häiriöiden vaikutuksen pienentämiseksi on kehitetty erilaisia koodausmenetelmiä, joilla signaalia pyritään suojaamaan häiriöiltä ja joiden avulla pyritään myös poistamaan häiriöiden aiheuttamia virheitä signaalissa. Eräs paljon käytetty koodausmenetelmä on konvoluutiokoodaus. Konvoluutiokoodauksessa lähetettävä symboleista koostuva signaali koodataan koodisanoiksi, jotka perustuvat lähetettävien symbolien konvoluutioon niin sanottujen generaattoripolynomien (koodauspolynomien) kanssa. Konvoluutiokoodin määrittävät koodaussuhde sekä koodauspolynomit. Koodaussuhde ( $k/n$ ) tarkoittaa tuotettujen koodattujen symbolien lukumäärää ( $n$ ) suhteessa koodattavien symbolien lukumäärään ( $k$ ). Kooderi toteutetaan usein siirtorekisterien avulla. Koodin vaikutussyvyydellä (constraint length  $K$ ) tarkoitetaan usein siirtorekisterin pituutta. Kooderia voidaan pitää tilakoneena, jossa on  $2K-1$  tilaa.

Vastaanottimessa dekodataan kanavassa edennyt koodattu signaali. Konvoluutiokoodi dekodataan yleensä käyttäen apuna trellistä, jonka solmut kuvaavat signaalin koodauksessa käytetyn enkooderin tiloja, ja trelliksen eri vaiheisiin kuuluvien solmujen väliset polut kuvaavat sallitut tilasiirtymät. Dekooderissa on pyrkimyksenä selvittää kooderin peräkkäiset tilat eli siirtymät tiloista toiseen. Siirtymien selvittämiseksi dekodерissa lasketaan metriikoita, joita on kahdenlaisia: polkumetriikat (path metrics, myös state metrics) ja siirtymämetriikat (branch metrics). Polkumetriikka edustaa todennäköisyyttä sille, että vastaanotetun signaalin sisältämien symbolien joukko johtaa kyseisen solmun kuvaamaan tilaan. Siirtymämetriikka edustaa eri siirtymien todennäköisyyksiä.

Trelliksen laskennan toteutuksessa peruselementti on ns. ACS-yksikkö (Add, Compare, Select), jonka avulla solmuissa lasketaan polkumetriikka edellisten solmujen polkumetriikoiden ja solmujen välisten siirtymämetriikoiden avulla. Laitteistototeutuksen tehokkuuden kannalta kaksi tällaista ACS-yksikköä yhdistetään usein kaksoisyksiköksi (twin-unit), jota kutsutaan perhoseksi (butterfly). Kukin yksikkö pystyy siis laskemaan kaksi solmua samanaikaisesti.

Konvoluutiokoodin dekodauksessa käytetään yleisesti Viterbi-algoritmia. Viterbi-algoritmi on laskennallisesti vaativa tehtävä. Yleinen ongelma Viterbi-algoritmissa on se, että kun konvoluutiokoodin koodaussyvyys on suuri (esimerkiksi 9, kuten UTMS-järjestelmän WCDMA:ssa), täytyy Viterbi-algoritmissa käydä yhden bitin dekodaukseksi lävitse  $2^{(9-1)}$  eli 256 tilaa. Jotta tällaisen trelliksen laskenta tapahtuisi riittävällä nopeudella, täytyy useilla ACS-yksiköillä tapahtuvaa laskentaa suorittaa rinnakkaisesti. Yleisesti rinnakkaisten kaksois-ACS-yksiköiden muodostamaa joukkoa sanotaan pankiksi. Tyypillisesti rinnakkaisia ACS-yksiköitä on vähemmän kuin trelliksessä on tiloja. Tästä syystä täytyy laskettuja metriikoita tallentaa väliaikaismuistiin. Tunnetun tekniikan mukaisissa toteutuksissa kutakin uutta laskettua metriikkaa kohden on yksi muistista luku- ja muistiinkirjoitusoperaatio. Tästä syystä muistien koko on suuri ja dataväylät, joilla tietoa siirretään muistien ja ACS-yksiköiden välillä on suuri. Tämä puolestaan huonontaa integrointiheyttä toteutettaessa laskenta kuvon avulla. Esimerkiksi jos toteutus käsittää 32 rinnakkaista ACS-yksikköä, ja polkumetriikka esitetään 12 bittisinä lukuina, muistin ja dataväylän koko on 384 bittiä leveä.

Kun laskenta suoritetaan ohjelmistollisesti, dekooderiohjelmiston suorituskyyä heikentää suuri muistin käyttö, joka on oleellisesti hitaampaa kuin prosessorin rekisterien käyttö. Rekistereitä prosessoreissa on rajallinen määrä.

### **Keksinnön lyhyt selostus**

Keksinnön tavoitteena on toteuttaa menetelmä ja järjestely dekodauksen toteuttamiseksi edullisesti. Tämä saavutetaan menetelmällä Viterbi-dekodauksen suorittamiseksi trelliksen avulla, jossa lasketaan joukko trelliksen peräkkäisiä metriikoita käyttämällä yhdellä tai useammalla ACS-yksiköllä lasketun trelliksen edellisen vaiheen metriikoita suoraan trelliksen seuraavan vaiheen laskennassa käytettävien yhden tai useamman ACS-yksikön sisäänmenona.

Keksinnön kohteena on myös järjestely Viterbi-dekoodauksen suorittamiseksi trelliksen avulla, joka järjestely käsittää yhden tai useamman ACS-yksikön. ACS-yksiköiden sisäänmenojen ja ulostulojen väliset kytkennät on toteutettu siten, että trelliksen peräkkäisten metriikoiden laskennassa ACS-yksiköiden ulostuloista saatavat trelliksen edellisen vaiheen lasketut metriikat on  
 5 kytketty suoraan trelliksen seuraavan vaiheen laskennassa käytettävien ACS-yksiköiden sisäänmenoihin.

Keksinnön edullisia toteutusmuotoja on kuvattu epäitsenäisissä patenttivaatimuksissa. Eräässä toteutusmuodossa valitaan ACS-yksiköiden sisäänmenoparit ja ryhmitellään kaksoisyksiköiden väliset kytkennät sillä tavoin, että metriikoiden tallentaminen muistiin laskennan aikana minimoituu. Koska  
 10 välimuistin käyttö on hidasta, saadaan ratkaisulla esimerkiksi se etu, että laskenta voidaan suorittaa nopeasti. Ratkaisu soveltuu sekä laite- että ohjelmistototeutukseen.

Edullisten toteutusmuotojen mukaisia ratkaisuja voidaan hyödyntää kaikissa sovelluksissa, joissa käytetään trellistä. Tällaisia sovelluksia ovat paitsi yllä mainittu konvoluutiokoodin dekoodaus myös muunlaisten koodien dekoodaus, ekvalisointi, monen käyttäjän ilmaisumenetelmät (multiuser decoding), ja puheentunnistus  
 15

## 20 Kuvioluettelo

Keksintöä selostetaan nyt lähemmin edullisten suoritusmuotojen yhteydessä, viitaten oheisiin piirroksiin, joissa

kuvio 1 esittää yksinkertaistettua lohkokaaaviota radiojärjestelmien rakenteesta,

25 kuviot 2A ja 2B esittävät erästä ACS-yksikköä,  
 kuviot 3A ja 3B esittää ACS-yksiköiden välisiä kytkentöjä,  
 kuviot 4A – 4C havainnollistavat esimerkkejä toteutusmuodoista,  
 kuvio 5 esittää esimerkkiä ohjelmistototeutuksesta ja  
 kuvio 6 havainnollistaa keksintöä vuokaavion avulla.

## 30 Toteutusmuotojen kuvaus

Kuvio 1 esittää esimerkkiä tiedonsiirtojärjestelmästä, jossa edullisia toteutusmuotoja voidaan soveltaa. Eräs kysymykseen tuleva järjestelmä on esimerkiksi UMTS (Universal Mobile Telecommunication System), ja muut laajakaistaiset järjestelmät. Jatkossa edullisia toteutusmuotoja kuvataan käyttä-

mällä esimerkkinä UMTS-järjestelmää, siihen kuitenkin rajoittumatta, kuten alan ammattimiehelle on selvää.

Selvennetään hiukan hakemuksessa käytettävää terminologiaa. Radiojärjestelmällä tarkoitetaan tässä yhteydessä tietoliikennejärjestelmien radiokerrosta, RAT (Radio Access Technology), joka kuuluu ns. AS:ään (Access Stratum), jonka yläpuolella tietoliikennejärjestelmät käsittävät radioriippumattoman kerroksen NAS (Non Access Stratum), joka käyttää erillisten radiojärjestelmien palveluita.

Tarkastellaan kuviota 1, joka havainnollistaa radiojärjestelmien rakennetta. Kuvio 1 on yksinkertaistettu lohkokaavio, joka kuvaa verkkoelementtien tasolla radiojärjestelmien tärkeimmät osat sekä niiden väliset rajapinnat. Verkkoelementtien rakennetta ja toimintoja ei kuvata kovin tarkasti, koska ne ovat yleisesti tunnettuja.

Kuviossa 1 tietoliikennejärjestelmän radioriippumatonta kerrosta kuvaa runkoverkko CN (Core Network) 100. Radiojärjestelmiä kuvaavat ensimmäinen radiojärjestelmä, eli radioliityntäverkko 130 ja toinen radiojärjestelmä eli tukiasemajärjestelmä BSS (Base Station System) 160. Lisäksi kuviossa on esitetty tilaajapäätelaitte UE (User Equipment) 170. Termi UTRAN tulee sanoista UMTS Terrestrial Radio Access Network, eli radioliityntäverkko 130 on toteutettu laajakaistaisella koodijakoisella monikäyttötekniikalla (Wideband Code Division Multiple Access, WCDMA). Tukiasemajärjestelmä 160 on toteutettu aikajakoisella monikäyttötekniikalla (Time Division Multiple Access, TDMA).

Yleisellä tasolla voidaan myös esittää sellainen määrittely, että radiojärjestelmä muodostuu tilaajapäätelaitteesta, joka tunnetaan myös esimerkiksi nimillä käyttäjälaite ja matkapuhelin, ja verkko-osasta, joka sisältää radiojärjestelmän kiinteän infrastruktuurin radioliityntäverkon tai tukiasemajärjestelmän.

Runkoverkon 100 rakenne vastaa yhdistettyä GSM- ja GPRS-järjestelmien rakennetta. GSM-verkkoelementit vastaavat piirikytkentäisten yhteyksien toteuttamisesta, ja GPRS-verkkoelementit pakettikytkentäisten yhteyksien toteuttamisesta, osa verkkoelementeistä sisältyy kuitenkin molempiin järjestelmiin.

Matkapuhelinkeskus MSC (Mobile Services Switching Centre) 102 on runkoverkon 100 piirikytkentäpuolen keskipiste. Samaa matkapuhelinkeskusta 102 voidaan käyttää palvelemaan sekä radioliityntäverkon 130 että tukiasemajärjestelmän 160 yhteyksiä. Matkapuhelinkeskuksen 102 tehtäviin kuu-

luu: yhteyksien välitys (switching), haku (paging), käyttäjälaitteen sijaintipaikan rekisteröinti (location registration), kanavanvaihdon hallinta (handover management), tilaajan (subscriber) laskutustietojen (billing information) keruu, tiedon salausparametrin hallinta (encryption parameter management), taajuusallo-

5 koinnin hallinta (frequency allocation management) ja kaiunpoisto (echo cancellation). Matkapuhelinkeskuksien 102 lukumäärä voi vaihdella: pienellä verkko-operaattorilla voi olla vain yksi matkapuhelinkeskus 102, mutta suurissa runkoverkoissa 100 niitä voi olla useampia.

Suurissa runkoverkoissa 100 voi olla erillinen yhdyskeskus GMSC

10 (Gateway Mobile Service Switching Centre) 110, joka hoitaa runkoverkon 100 ja ulkopuolisten verkkojen 180 väliset piirikytkentäiset yhteydet. Yhdyskeskus 110 sijaitsee matkapuhelinkeskuksien 102 ja ulkopuolisten verkkojen 180 välissä. Ulkopuolinen verkko 180 voi olla esimerkiksi yleinen matkaviestinverkko PLMN (Public Land Mobile Network) tai yleinen puhelinverkko PSTN (Public

15 Switched Telephone Network).

Kotirekisteri HLR (Home Location Register) 114 sisältää pysyvän tilaajarekisterin, eli esimerkiksi seuraavat tiedot: kansainvälisen matkaviestintilaajan tunnuksen (International Mobile Subscriber Identity, IMSI), matkaviestintilaajan kansainvälisen ISDN-numeron (Mobile Subscriber ISDN Number, MSISDN), autentikointiavaimen (Authentication Key), ja radiojärjestelmän tukissa GPRS:ää PDP-osoitteen (PDP = Packet Data Protocol).

20

Vierailijarekisteri VLR (Visitor Location Register) 104 sisältää sijainnin seurantaa (roaming) koskevaa informaatiota matkapuhelinkeskuksen 102 alueella olevista käyttäjälaitteista 170. Vierailijarekisteri 104 sisältää aika pitkälti samaa informaatiota kuin kotirekisteri 114, mutta vierailijarekisterissä 104 kyseinen informaatio on vain tilapäisesti.

25

Tunnistuskeskus AuC (Authentication Center) 116 sijaitsee fyysisesti aina samassa paikassa kuin kotirekisteri 114, ja se sisältää tilaajan tunnistusavaimen (Individual Subscriber Authentication Key) Ki, CK (Ciphering Key)

30 sekä vastaavan IMSI:n.

Kuviossa 1 kuvattavat verkkoelementit ovat toiminnollisia kokonaisuuksia, joiden fyysinen toteutus voi vaihdella. Tavallisesti matkapuhelinkeskus 102 ja vierailijarekisteri 104 muodostavat yhden fyysisen laitteen, ja kotirekisteri 114 sekä tunnistuskeskus 116 toisen fyysisen laitteen.

35

Operointisolmu SGSN (Serving GPRS Support Node) 118 on runkoverkon 100 pakettikytkentäpuolen keskipiste. Operointisolmun 118 pääte-

tävä on lähettää ja vastaanottaa paketteja pakettikytkentäistä siirtoa tukevan käyttäjälaitteen 170 kanssa radioliityntäverkkoa 130 tai tukiasemajärjestelmää 160 käyttäen. Operointisolmu 118 sisältää käyttäjälaitetta 170 koskevaa tilaaja-tietoa sekä sijaintitietoa.

- 5 Yhdyskäytäväsolmu GGSN (Gateway GPRS Support Node) 120 on pakettikytkentäpuolen vastine piirikytkentäpuolen yhdyskeskukselle 110, kuitenkin sillä erotuksella, että yhdyskäytäväsolmun 120 on kyettävä reitittämään myös runkoverkosta 100 ulkopuolisiin verkkoihin 182 ulosmenevän liikenteen, kun taas yhdyskeskus 110 reitittää vain sisään tulevan liikenteen. Esimerkissämme ulkopuolisia verkkoja 182 edustaa Internet.

- 10 Ensimmäinen radiojärjestelmä eli radioliityntäverkko 130 muodostuu radioverkon alijärjestelmistä RNS (Radio Network Subsystem) 140, 150. Kukin radioverkon alijärjestelmä 140, 150 muodostuu radioverkko-ohjaimista RNC (Radio Network Controller) 146, 156 sekä B-solmuista 142, 144, 152, 154. B-solmu on melko abstrakti käsite, ja usein sen sijasta käytetään termiä tukiasema.

- Radioverkko-ohjain 146 kontrolloi alaisuudessaan olevia B-solmuja 142, 144. Periaatteessa pyritään siihen, että radiotien toteuttavat laitteet niihin liittyvine toimintoineen sijaitsevat B-solmuissa 142, 144, ja hallintalaitteet sijaitsevat radioverkko-ohjaimessa 146.

- Radioverkko-ohjain 146 hoitaa esimerkiksi seuraavat tehtävät: B-solmun 142, 144 radioresurssien hallinta, solujenväliset kanavanvaihdot (inter-cell handover), taajuushallinta eli taajuuksien allokointi B-solmuille 142, 144, taajuushyppelysekvenssien hallinta, aikaviiveiden mittaaminen nousevalla siirtotielä, käytönohjauksen (operation and maintenance) rajapinnan toteutus ja tehonsäädön hallinta.

- B-solmu 142, 144 sisältää yhden tai useamman lähetinvastaanottimen, joilla toteutetaan WCDMA-radorajapinta. Tyypillisesti B-solmu palvelee yhtä solua, mutta myös sellainen ratkaisu on mahdollinen, jossa B-solmu palvelee useaa sektoroitua solua. Solun läpimitta voi vaihdella muutamista metreistä kymmeneen kilometriin. B-solmulle 142, 144 kuuluvat esimerkiksi seuraavat tehtävät: TA:n (timing advance) laskeminen, nousevan siirtotien mittaukset, kanavakoodaus, koodaus salakieliseksi (encryption), salauksenpurku (decryption) ja taajuushyppely.

- 35 Toinen radiojärjestelmä eli tukiasemajärjestelmä 160 muodostuu tukiasemaohjaimesta BSC (Base Station Controller) 166 sekä tukiasemista BTS

(Base Transceiver Station) 162, 164. Tukiasemaohjain 166 kontrolloi tukiasemaa 162, 164. Periaatteessa pyritään siihen, että radiotien toteuttavat laitteet niihin liittyvine toimintoineen sijaitsevat tukiasemassa 162, 164, ja hallintalaitteet sijaitsevat tukiasemaohjaimessa 166. Tukiasemaohjain 166 hoitaa oleellisesti samoja tehtäviä kuin radioverkko-ohjain.

Tukiasema 162, 164 sisältää ainakin yhden lähetinvastaanottimen, joka toteuttaa yhden kanta-aallon, eli kahdeksan aikaväliä, eli kahdeksan fyysistä kanavaa. Tyypillisesti yksi tukiasema 162, 164 palvelee yhtä solua, mutta myös sellainen ratkaisu on mahdollinen, jossa yksi tukiasema 162, 164 palvelee useaa sektoroitua solua. Tukiasemaan 162, 164 katsotaan kuuluvan myös transkooderin, jolla suoritetaan muunnos radiojärjestelmässä käytetyn puheenkoodausmuodon ja yleisessä puhelinverkossa käytetyn puheenkoodausmuodon välillä. Käytännössä transkooderi kuitenkin fyysisesti yleensä sijaitsee matkapuhelinkeskuksessa 102. Tukiasemalle 162, 164 kuuluvat vastaavat tehtävät kuin B-solmulle.

Tilaajapäätelaite 170 koostuu kahdesta osasta: matkaviestinlaite (Mobile Equipment, ME) 172 ja UMTS-tilaajan tunnistusyksikkö (UMTS Subscriber Identity Module, USIM) 174. Tilaajapäätelaite 170 sisältää ainakin yhden lähetinvastaanottimen, jolla toteutetaan radioyhteys radioliityntäverkkoon 130 tai tukiasemajärjestelmään 160. Tilaajapäätelaite 170 voi sisältää ainakin kaksi erilaista tilaajan tunnistusyksikköä. Lisäksi tilaajapäätelaite 170 sisältää antennin, käyttöliittymän sekä akun. Nykyisin tilaajapäätelaitteita 170 on monenlaisia, esimerkiksi autoon asennettuja sekä kannettavia.

USIM 174 sisältää käyttäjään liittyvää tietoa, sekä erityisesti tietoturvallisuuteen liittyvää tietoa, esimerkiksi salausalgoritmin.

Tarkastellaan seuraavaksi Viterbi-dekoodauksen suorittamista. Viterbi-dekoodauksessa eräs perustoimenpide on ns lisää-vertaa-valitse (Add-Compare-Select, ACS) operaatio. Yksinkertaistaen esitettynä Viterbi-algoritmia laskettaessa lisätään aiempaan polkumetriikkaan siirtymämetriikan arvo, suoritetaan vertailu saatujen arvojen välillä ja valitaan suurin arvo uudeksi polkumetriikaksi. Laitteistototeutuksen kannalta on edullista toteuttaa ACS-operaatio pareittain. Tällöin saadaan yksikkö, joka suorittaa siis kaksi ACS-operaatiota, ja jolla on kaksi yhteistä sisäänmenoa ja kaksi ulostuloa. Tällaista yksikköä sanotaan alan kirjallisuudessa perhoseksi (butterfly).

ACS-operaatiot voidaan suorittaa joko laitteistototeutuksena tai ohjelmistollisesti. Laitteistototeutuksena keksinnön edullisissa toteutusmuodoissa



valitaan yksiköiden väliset kytkennät siten, että joukko trelliksen peräkkäisiä metriikoita lasketaan tallentamatta metriikka-arvoja välillä muistiin eli käyttämällä edellisen trelliksen vaiheen laskettuja metriikoita suoraan seuraavan trelliksen vaiheen laskennan sisäänmenona. Ohjelmistototeutuksena vastaava ratkaisu vaikuttaa sisään- ja ulostulodatojen välisiin tallennustapoihin.

Kuviossa 2A on eräs esimerkki tällaisen kaksois-ACS-yksikön 200 toteutuksesta. Sisäänmenona 202, 204 on edellisen sarakkeen kahden solmun polkumetriikat. Ulostulossa 206, 208 on seuraavan sarakkeen kaksi uutta polkumetriikkaa. Sisäänmenona 210 on myös siirtymämetriikat. Yksikkö käsittää neljä summaina 212 – 218, joissa polkumetriikat ja siirtymämetriikat summataan. Summainen ulostulot viedään vertailijoille 220, 222, joissa tutkitaan mitkä summista ovat suurimmat tai pienimmät, riippuen toteutustavasta. Vertailijoilta lähtee päätössignaalit multipleksereille 224, 226, jotka ohjaavat valitut summat ulostuloon, seuraavan sarakkeen polkumetriikoiksi. Päätössignaalit voidaan myös viedä yksiköstä ulostulona 228. Kuviossa 2B on yksinkertaistettu kaksoisyksikön merkintätapa. Selkeyden vuoksi siirtymämetriikoita ei kuviossa esitetä.

Eräässä toteutusmuodossa valitaan kaksoisyksiköiden sisäänmenoparit ja ryhmitellään kaksoisyksiköiden väliset kytkennät sillä tavoin, että välimuistin tarve minimoituu. Kaksoisyksiköiden ryhmittely ja kytkennät valitaan siten, että kaksoisyksikön ulostulossa olevat polkumetriikat voidaan käyttää välittömästi seuraavan kaksoisyksiköiden sisäänmenosignaalina. Tällöin ei ulostuloissa olevia polkumetriikoita tarvitse tallettaa välimuistiin. Tarkastellaan ensin binäärisiä trelliksiä, joissa yhteen tilaan pääsee kahdesta tilasta ja vastaavasti yhdestä tilasta pääsee kahteen tilaan. Jos datapolkujen lukumäärä on  $P$ , on mahdollista laskea  $\log_2 P$  polkumetriikkaa yhtä muistiin luku/kirjoitusoperaatiota kohti. Tämä seuraa siitä, että, jos ollaan tietyssä tilassa, niin siihen on voitu tulla kahdesta tilasta edellisellä sarakkeella, joihin molempiin on voitu tulla kahdesta tilasta sitä edellisellä sarakkeella, jne. Eli  $x$ :n askeleen takaa on voitu tulla  $2^x$  eri tilasta. Esitetyssä ratkaisussa on kytkennät järjestetty siten, että  $x$ :n askeleen takaa ei lasketa yhtään enempää kuin  $2^x$  tilaa, jotka tarvitaan ko. tilan laskemiseen. Toisin sanoen  $x = \log_2 P$  ja  $2^x = P$ . Edullisesti kytkennät voidaan valita siten, että sama kytkentä voidaan soveltaa useiden sarakkeiden laskennassa.

Muissa kuin binäärisissä trelliksissä, eli siis trelliksissä joissa yhteen tilaan pääsee useammasta kuin kahdesta tilasta ja vastaavasti yhdestä tilasta

pääsee useampaan kuin kahteen tilaan. Tällöin edellä mainittu  $\log_2 P$  muuttuu muotoon  $\log_Y P$ , missä  $Y$  on se luku, kuinka monta yhteyttä yhdestä tilasta lähtee/tulee.

Tarkastellaan kuviota 3A, jossa havainnollistetaan kaksoisyksiköiden välisiä kytkentöjä. Kuviossa on esitetty 16-tilaisen koodin trelliksen laskentaa. Kussakin sarakkeessa on siis 16 tilaa, jotka on numeroitu 0 – 15. Kuviossa on esitetty neljä trelliksen saraketta 300, 302A, 304A ja 306A. Toteutuksessa käytetään kahta neljän kaksoisyksikön pankkia. Täten siis datapolkujen lukumäärä on kahdeksan. Välimuistista luettavat ensimmäisen sarakkeen 300 polkumetriikat ovat ensimmäisenä sisäänmenona kahdelle kaksoisyksikköpankille 308A ja 308B. Kaksoisyksiköiden ulostulossa ovat seuraavan sarakkeen 302A polkumetriikat. Kuviossa on edelleen esitetty sama sarake uudelleen ryhmiteltynä 302B. Sarakkeiden 302A ja 302B väliset viivat kuvaavat uudelleenryhmitelyä. Ryhmittelyn avulla ohjataan kaksoisyksiköiden 308A ja 308B ulostulossa olevat polkumetriikat seuraavien kaksoisyksiköiden 310A ja 310B sisäänmenoihin siten, että välimuistia ei tarvita. Vastaavasti suoritetaan kaksoisyksiköiden 310A ja 310B ulostulojen 304A ryhmittely muotoon 304B kaksoisyksiköiden 312A ja 312B sisäänmenoihin. Kaksoisyksiköiden 312A ja 312B ulostulossa olevat metriikat 306A voidaan myös ryhmitellä muotoon 306B, jonka jälkeen ne talletetaan välimuistiin. Koska polkuja kuvion 3A esimerkissä on 8, voidaan kahdeksaa luku/kirjoitus-operaatiota kohti suorittaa  $8 \cdot \log_2(8) = 24$  polkumetriikan laskentaa. Lukuoperaatiot suoritetaan siis metriikoille 300 ja kirjoitusoperaatiot metriikoille 306B.

Kuten kuviosta 3A huomataan, ryhmittely on mahdollista suorittaa siten, että viivoitukset ovat samoja eri sarakkeiden välillä. Tämä helpottaa laitteiston toteutusta. Toteutuksessa voidaan esimerkiksi hyödyntää neljän kaksoisyksikön pankkeja, kuten kuvio 3A esittää. Kaksoisyksikköpankki ja sitä seuraava ryhmittely toistuvat samankaltaisina kuvion 3A esimerkissä kuusi kertaa, joten samoja pankkeja voidaan hyödyntää laskennan eri vaiheissa.

Kuviossa 3B havainnollistetaan toista esimerkkiä. Kuviossa on osa 256-tilaisesta koodista. ajan hetkellä  $n-2$  luetaan muistista polkumetriikat 314, jotka viedään kaksoisyksikköpankille 316. Lasketut metriikat viedään edelleen kaksoisyksiköille 318 ja 320, jonka ulostulosta saadaan siis metriikat 322, jotka kirjoitetaan muistiin. Vaiheissa 324 polkumetriikat prosessoidaan paikallisesti eikä muistiin lukua tai kirjoitusta siis tarvita.

Trelliksen laskenta kaksoisyksiköiden avulla ja kytkennät yksiköiden välillä voidaan toteuttaa usealla eri tavalla. Tarkastellaan kuvioita 4A- 4C. Kuviossa 4A on esitetty muisti 400, sekä kolme kaksoisyksikköpankkia 402, 404, 406 peräkkäiskytkennässä. Edellisessä pankissa lasketut metriikat voidaan  
 5 viedä suoraan seuraavalle pankille sisäänmenoihin ilman välitallennusta muistiin. Esimerkiksi jos datalinjojen lukumäärä on kahdeksan ja kussakin pankissa siis neljä kaksoisyksikköä, voidaan peräkkäin kytkeä edullisesti kolme pankkia, ja kolmannen pankin 406 ulostulot viedään muistiin.

Tarkastellaan kuvion 3A esimerkin toteutusta kuvion 4A järjestelyn  
 10 avulla. Ensimmäisessä vaiheessa pankki 402 lukee muistista 400 sisäänmenoihinsa polkumetriikoiden 300 kahdeksan ylintä arvoa, ja suorittaa laskennan. Tulokset, eli metriikoiden 302A kahdeksan ylintä arvoa viedään pankille 404 sisäänmenoksi. Samalla pankki 403 lukee sisäänmenoonsa muistista 400 polkumetriikoiden 300 kahdeksan alinta arvoa. Pankit 402 ja 404 suorittavat  
 15 laskennan, ja pankin 404 tulos (polkumetriikoiden 304A kahdeksan ylintä arvoa) viedään pankin 406 sisäänmenoksi, ja vastaavasti pankin 402 tulos (polkumetriikoiden 302A kahdeksan alinta arvoa) viedään pankin 404 sisäänmenoksi. Tässä vaiheessa pankki 402 ei laske mitään. Pankit 404 ja 406 suorittavat laskennan, ja pankin 406 tulos (polkumetriikoiden 306A kahdeksan  
 20 alinta arvoa) viedään muistiin 400. Pankin 404 tulos (polkumetriikoiden 304A kahdeksan alinta arvoa) viedään pankin 404 sisäänmenoksi. Lopuksi pankki 406 suorittaa viimeisen laskennan ja pankin 406 tulos (polkumetriikoiden 306A kahdeksan ylintä arvoa) viedään muistiin 400. Trelliksen laskentaa voidaan ketjuttaa siten, ettei pankkien joutoaikaa tapahdu.

25 Kuviossa 4B on esitetty muisti 400, sekä kolme kaksoisyksikköpankkia 402, 404, 406 rinnakkaisytkennässä. Järjestely käsittää kytkimet 408, 410, joilla ohjataan polkumetriikoiden vientiä muistin ja pankkien välillä.

Kuviossa 4C on esitetty muisti 400 sekä yksi kaksoisyksikköpankki 402. Järjestely käsittää kytkimen 412, joilla ohjataan polkumetriikoiden vientiä  
 30 muistin ja pankin välillä. Esimerkiksi kuvion 3A laskentaa toteutettaessa kuvion 4C järjestelyn avulla voidaan polkumetriikoita viedä pankin ulostulosta suoraan takaisin pankin sisäänmenoon. Aluksi kytkin 412 on ala-asennossa ja pankki 402 lukee muistista sisäänmenoihinsa polkumetriikoiden 300 kahdeksan ylintä arvoa, ja suorittaa laskennan. Kytkin käännetään yläasentoon, ja tulokset, eli  
 35 metriikoiden 302A kahdeksan ylintä arvoa viedään uudestaan pankin 402 sisäänmenoon. Pankki 402 suorittaa laskennan, ja tulos (polkumetriikoiden 304A

kahdeksan ylintä arvoa) viedään uudestaan pankin 402 sisäänmenoon. Pankki 402 suorittaa laskennan vielä kerran, ja tulos (polkumetriikoiden 306A kahdeksan ylintä arvoa) viedään muistiin 400. Kytkin käännetään ala-asentoon ja pankki voi lukea uudet polkumetriikat muistista 400.

- 5 Kuviossa 5 havainnollistetaan esimerkkiä, jossa laskenta suoritetaan ohjelmallisesti. Kuviossa esitetään laskentayksikkö 500 (ALU, Arithmetic Logic Unit), joka vastaa laskentaoperaatioiden suorituksesta tyypillisessä prosessorissa. Laskentayksikkö saa laskennassa tarvitsemansa datan rekistereistä 502, jotka ovat nopealukuisia muistipaikkoja. rekistereiden lukumäärä on  
10 nykyisissäkin prosessoriratkaisuissa rajoitettu, joten niiden lisäksi prosessori käsittää muistin 504, johon voidaan tallentaa käsiteltyä dataa, kuten välituloksia, jotka eivät mahdu rekistereihin. Muistin 504 luku/kirjoitusoperaatiot ovat oleellisesti hitaampia kuin rekistereiden käsittely. Prosessori toimii kellopulssin ohjaamana siten, että yhden kellojakson aikana tyypillisesti suoritetaan yksi  
15 laskentatoimenpide tai rekisterin kirjoitus/lukuoperaatio. Muistin 504 kirjoitus/luku vie useita kellojaksoja. Kellopulssia ei ole näytetty kuviossa 5.

- Datan luku laskentayksiköltä 500 rekistereihin 502 tai datan siirto muistista 504 rekistereihin 502 tapahtuu multiplekserin 506 kautta. Jos esimerkiksi laskennan tulos viedään laskentayksiköstä 500 muistiin 504, se vie tyypil-  
20 lisesti enemmän kuin yhden kellojakson. Datan luku muistista 504 laskentayksikölle vie multiplekserin 506 ja rekistereiden 502 kautta kolme kellojaksoa. Laskennan tulos viedään laskentayksiköstä muistiin 504, se vie yhden kellojakson. Jos laskennan tulos viedään laskentayksiköstä rekistereihin 502, se vie yhden kellojakson.

- 25 Keksinnön edullisissa toteutusmuodoissa laskennan tuloksia ei tarvitse tallettaa muistiin 504, vaan dataa voidaan pitää rekistereissä, koska edellisen vaiheen ulostulodata toimii suoraan seuraavan vaiheen sisäänmenodata-na. Tällöin datan siirtoon kuluu vähemmän kellojaksoja ja laskenta täten siis nopeutuu. Jos 16 rekisteriä varataan polkumetriikoiden tallennukseen, muistin  
30 504 käyttö vähenee 75%. Jos 8 rekisteriä varataan polkumetriikoiden tallennukseen, muistin 504 käyttö vähenee 66%.

- ACS-yksiköiden väliset kytkennät ja/tai polkumetriikoiden ryhmittely voidaan tallettaa etukäteen muistiin tai tilakoneisiin. Toinen vaihtoehto on suorittaa kytkentöjen määrittely dekodauksen yhteydessä. Jos kytkentöjen mää-  
35 rittely tai polkumetriikoiden ryhmittely tehdään dekodauksen yhteydessä, tarvitaan silloin tilakone, joka antaa oikeat arvot oikealla hetkellä.

Tarkastellaan vielä kuvion 6 vuokaaviota, joka havainnollistaa erästä suoritustapaa. Vaiheessa 600 luetaan muistista joukko metriikoita sisäänmenoksi ACS-yksiköihin. Vaiheessa 602 suoritetaan trelliksen seuraavan vaiheen metriikoiden laskenta ACS-yksiköissä. Jos ei ole vielä tarvetta 604 tallentaa laskennan tuloksia muistiin, niin vaiheessa 606 viedään saadut metriikat ACS-yksiköiden ulostuloista ACS-yksiköiden sisäänmenoihin trelliksen seuraavan vaiheen laskentaa varten. Vaiheita 602 ja 606 toistetaan yksi tai useampi kertaa. Kun on tarve tallettaa laskennan tuloksia muistiin, niin tämä suoritetaan vaiheessa 608. Tarve tallentaa tulokset muistiin tapahtuu siis silloin, kun  $\log_2 P$  polkumetriikkaa yhtä muistiin luku/kirjoitusoperaatiota kohti on laskettu, missä  $P$  on rinnakkaisten datapolkujen lukumäärä.

Vaikka keksintöä on edellä selostettu viitaten oheisten piirustusten mukaiseen esimerkkiin, on selvää, ettei keksintö ole rajoittunut siihen, vaan sitä voidaan muunnella monin tavoin oheisten patenttivaatimusten puitteissa.

**Patenttivaatimuks t**

1. Järjestely Viterbi-dekoodauksen suorittamiseksi trelliksen avulla, joka järjestely käsittää yhden tai useamman ACS-yksikön (402 – 406), t u n n e t t u siitä, että ACS-yksiköiden (402 – 406) sisäänmenojen ja ulostulojen väliset kytkennät on toteutettu siten, että trelliksen peräkkäisten metriikoiden laskennassa ACS-yksiköiden ulostuloista saatavat trelliksen edellisen vaiheen lasketut metriikat on kytketty suoraan trelliksen seuraavan vaiheen laskennassa käytettävien ACS-yksiköiden sisäänmenoihin.

2. Patenttivaatimuksen 1 mukainen järjestely, t u n n e t t u siitä, että järjestely käsittää muistin (400) laskentatulosten tallentamista varten, ja että ACS-yksiköiden (402 – 406) sisäänmenojen ja ulostulojen väliset kytkennät on toteutettu siten, että yhtä muistiin luku/kirjoitusoperaatioparia kohti lasketaan log<sub>2</sub> P polkumetriikkaa, jossa P on datapolkujen lukumäärä ja Y on haarojen lukumäärä tilasta/tilaan.

3. Patenttivaatimuksen 1 mukainen järjestely, t u n n e t t u siitä, että ACS-yksiköt (402 – 406) on sovitettu laskemaan kukin Y ACS-operaatiota, missä Y on haarojen lukumäärä tilasta/tilaan.

4. Patenttivaatimuksen 1 mukainen järjestely, t u n n e t t u siitä, että järjestely käsittää useasta ACS-yksiköstä muodostuvia ACS-pankkeja.

5. Patenttivaatimuksen 4 mukainen järjestely, t u n n e t t u siitä, että järjestely käsittää ACS-pankkeja peräkkäiskytkenässä.

6. Patenttivaatimuksen 4 mukainen järjestely, t u n n e t t u siitä, että järjestely käsittää ACS-pankkeja rinnakkaiskytkennässä.

7. Patenttivaatimuksen 4 mukainen järjestely, t u n n e t t u siitä, että laskennassa käyttävän ACS-pankin ulostulo on takaisinkytkettävissä pankin sisäänmenoon.

8. Patenttivaatimuksen 4 mukainen järjestely, t u n n e t t u siitä, että ACS-pankki on toteutettu prosessorissa ohjelmallisesti.

9. Patenttivaatimuksen 9 mukainen järjestely, t u n n e t t u siitä, että prosessori on sovitettu tallentamaan trelliksen edellisen vaiheen laskettuja metriikoita rekisterimuistin, ja lukemaan talletetut metriikat rekisterimuistista laskettaessa trelliksen seuraavan vaiheen metriikoita.

10. Menetelmä Viterbi-dekoodauksen suorittamiseksi trelliksen avulla, t u n n e t t u siitä, että lasketaan joukko trelliksen peräkkäisiä metriikoita käyttämällä yhdellä tai useammalla ACS-yksiköllä (402 – 406) lasketun trellik-

sen edellisen vaiheen metriikoita suoraan trelliksen seuraavan vaiheen laskennassa käytettävien yhden tai useamman ACS-yksikön sisäänmenona.

11. Patenttivaatimuksen 10 mukainen menetelmä, tunnettu siitä, että laskentatuloksia talletetaan muistiin (400), ja että yhtä muistiin luku/kirjoitusoperaatioparia kohti lasketaan  $\log_Y P$  polkumetriikkaa, jossa P on datapolkujen lukumäärä ja Y on haarojen lukumäärä tilasta/tilaan.

12. Patenttivaatimuksen 10 mukainen menetelmä, tunnettu siitä, että suoritetaan laskenta useasta ACS-yksiköstä muodostuvilla ACS-pankillla.

10 13. Patenttivaatimuksen 12 mukainen menetelmä, tunnettu siitä, että kytketään laskennassa käytettävät ACS-pankit peräkkäiskytkentään.

14. Patenttivaatimuksen 12 mukainen menetelmä, tunnettu siitä, että kytketään laskennassa käytettävät ACS-pankit rinnakkaiskytkentään.

15 15. Patenttivaatimuksen 12 mukainen menetelmä, tunnettu siitä, että suoritetaan dekodaus ohjelmallisesti.

16. Patenttivaatimuksen 15 mukainen menetelmä, tunnettu siitä, että talletetaan lasketut metriikat rekisterimuistiin, ja luetaan trelliksen seuraavaa vaihetta laskettaessa edellisen vaiheen metriikat rekisteristä.

20 17. Patenttivaatimuksen 10 mukainen menetelmä, tunnettu siitä, että a) luetaan (600) muistista joukko metriikoita sisäänmenoksi ACS-yksiköihin,

b) suoritetaan (602) trelliksen seuraavan vaiheen metriikoiden laskenta ACS-yksiköissä,

25 c) viedään (606) saadut metriikat ACS-yksiköiden ulostuloista ACS-yksiköiden sisäänmenoihin trelliksen seuraavan vaiheen laskentaa varten, toistetaan askeleita b) ja c) yksi tai useampi kertaa.

18. Patenttivaatimuksen 17 mukainen menetelmä, tunnettu siitä, että toistetaan askeleita b) ja c) kunnes  $\log_Y P$  polkumetriikkaa on laskettu, jossa P on datapolkujen lukumäärä ja Y on haarojen lukumäärä tilasta/tilaan.

**(57) Tiivistelmä**

Keksinnön kohteena on menetelmä ja järjestely Viterbi-dekoodauksen suorittamiseksi trelliksen avulla. Järjestely käsittää yhden tai useamman ACS-yksikön (402 – 406). ACS-yksiköiden (402 – 406) sisäänmenojen ja ulostulojen väliset kytkennät on toteutettu siten, että trelliksen peräkkäisten metriikoiden laskennassa ACS-yksiköiden ulostuloista saatavat trelliksen edellisen vaiheen lasketut metriikat on kytketty suoraan trelliksen seuraavan vaiheen laskennassa käytettävien ACS-yksiköiden sisäänmenoihin.

(Kuvio 4A)



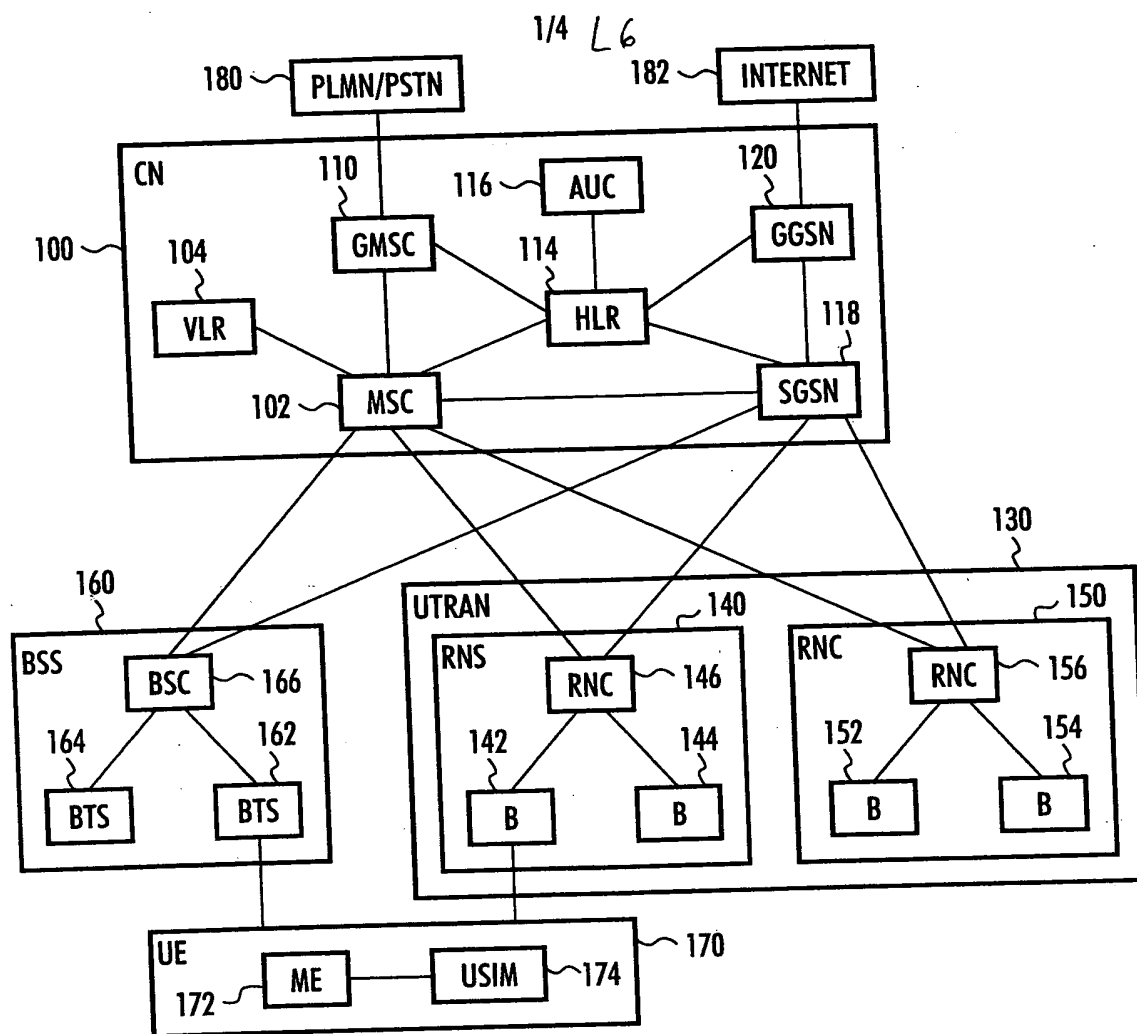


Fig. 1

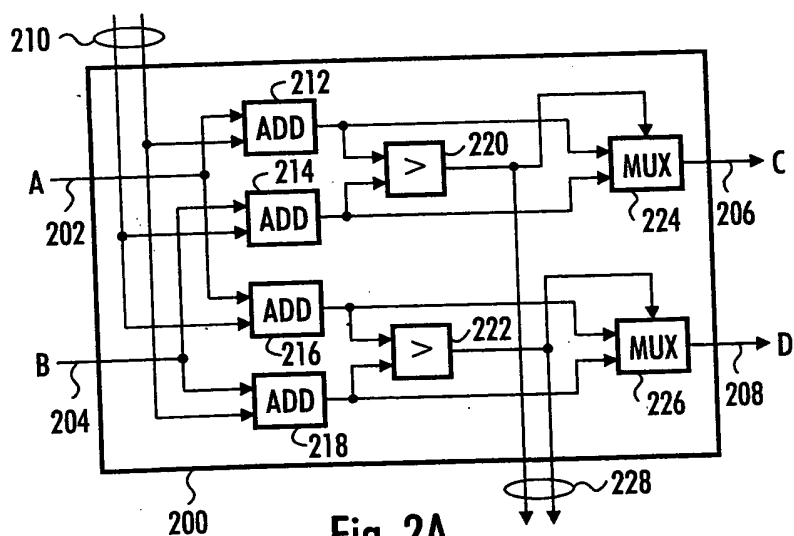


Fig. 2A

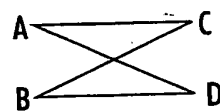


Fig. 2B

2/4 L6

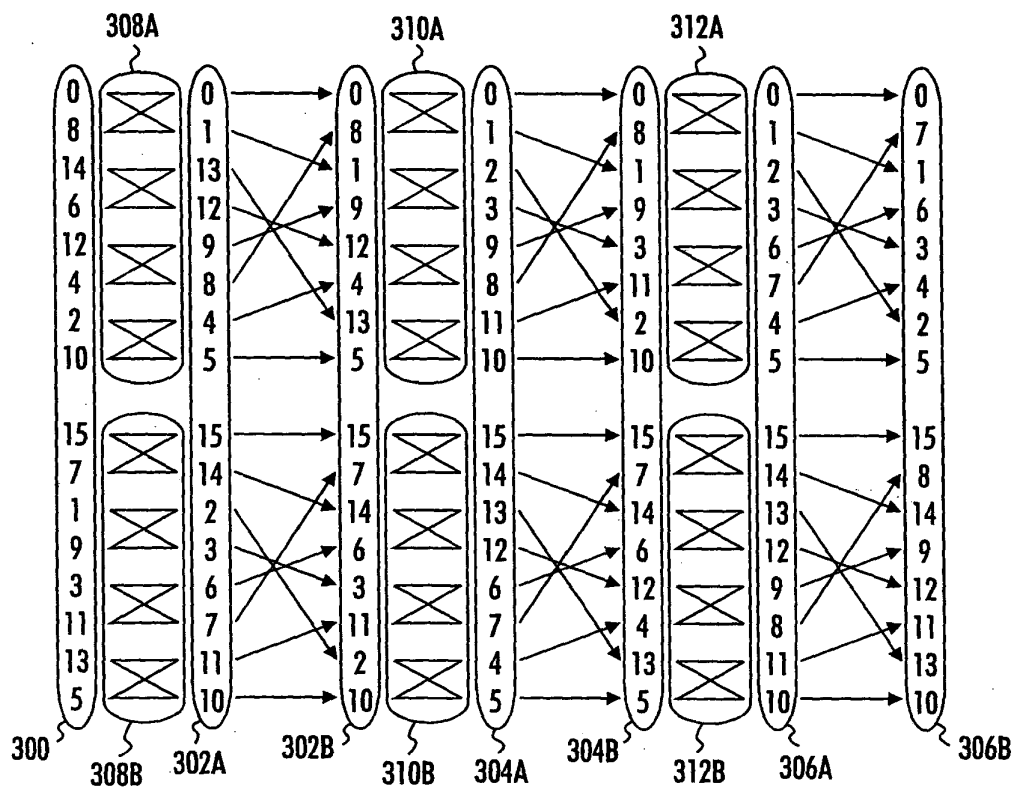


Fig. 3A

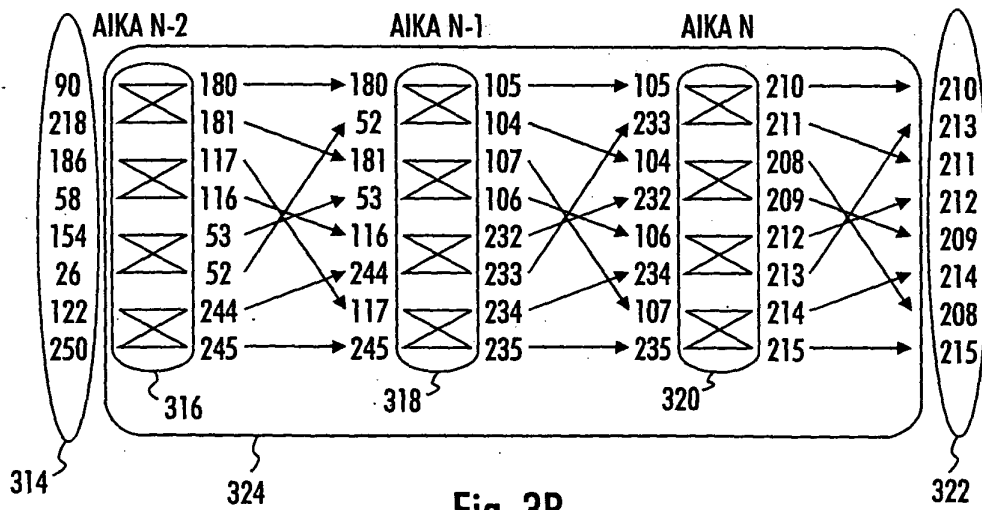


Fig. 3B

3/4 LG

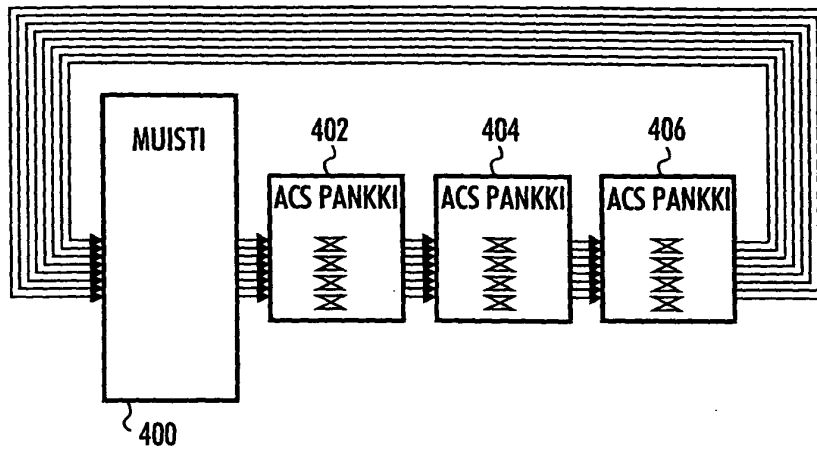


Fig. 4A

Fig. 4B

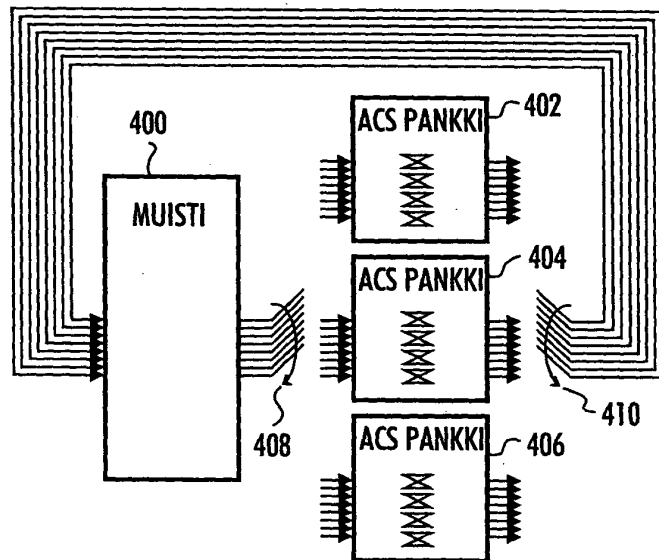
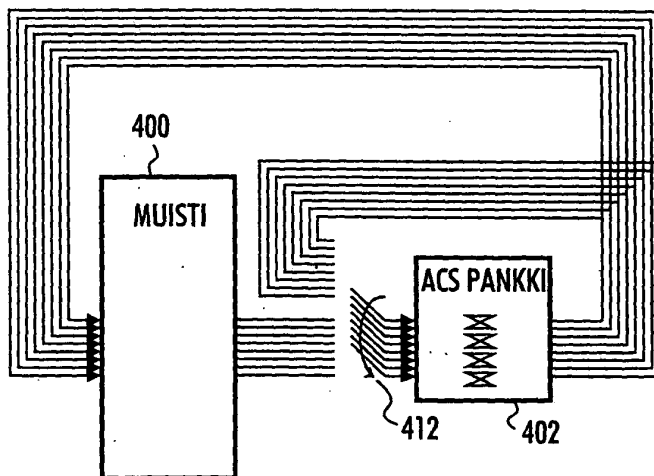


Fig. 4C



4/4 L6

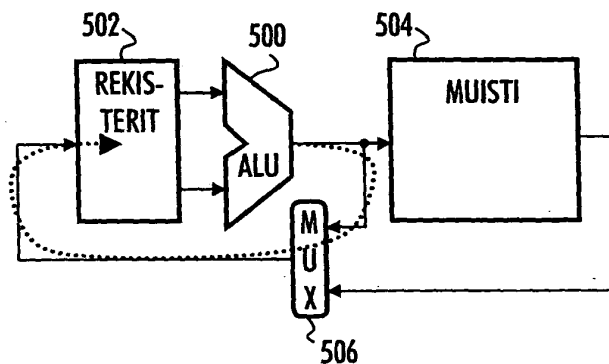


Fig. 5

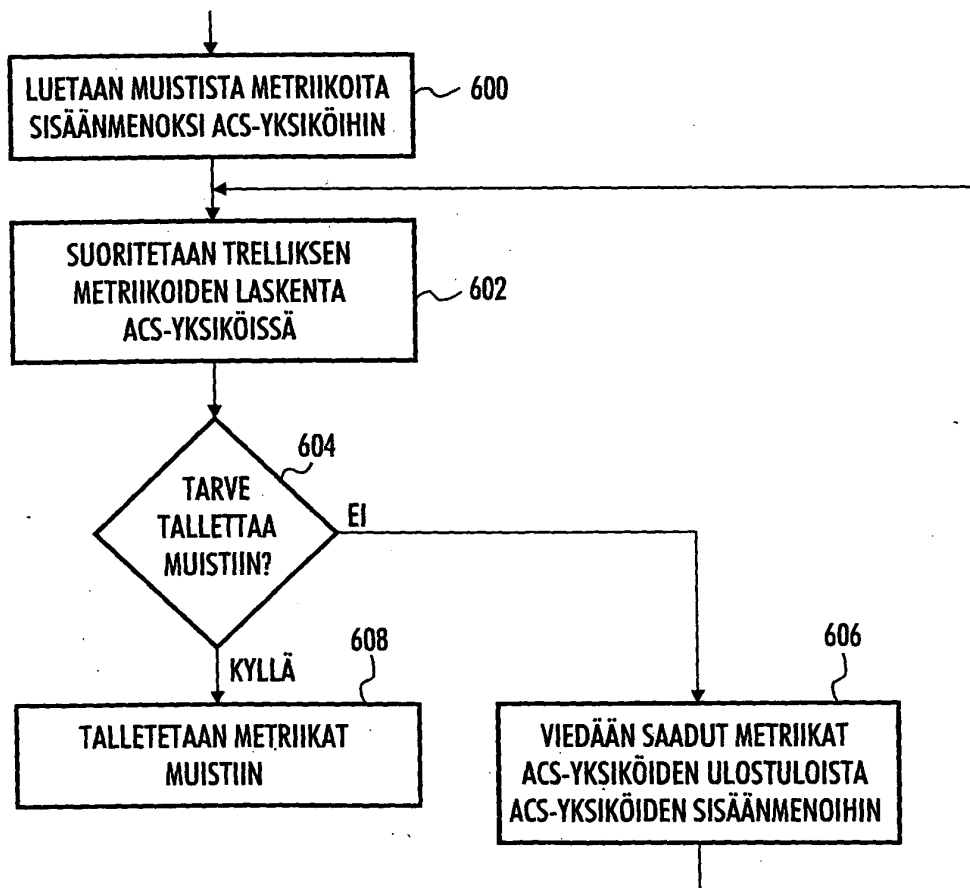


Fig. 6